Global Information Assurance Certification Paper

Steven Shields
Capital Sans
Security Essentials
13 February, 2001

"Web Server Folder Traversal" vulnerability (MS00-078)

On October 10, 2000, an anonymous user posted a message to the Packetstorm forum stating that by using a specific URL, he (or she) could execute the DIR command. Thus, the birth of the "Web Server Folder Traversal" vulnerability. Although an easy fix, this vulnerability is still in use today, and is likely the access method of choice for most attacks against Internet Information Servers (IIS).

Web Server Folder Traversal vulnerability will allow users to gain access to a computer system running Microsoft IIS versions 4.0 and 5.0. Using a malformed Uniform Resource Locator (URL), an individual could gain access to folders and files that are located on the server's logical drive. Gaining this type of access could result in a multitude of possibilities. Specifically, this access could allow an individual to add, change or delete data, run code already on the server or upload and run new code to the server.

A patch was created to correct this vulnerability in August 2000. Although not intended for this vulnerability, Microsoft posted a patch for the "File Permission Canonicalization" vulnerability that exhibited similar traits to this vulnerability. The File Permission Canonicalization vulnerability results when a canonicalization error affecting CGI scripts and ISAPI extensions occur. When an URL requesting either a CGI script or an ISAPI-mapped file - one located in a physical rather than a virtual folder - were malformed in a particular way, the wrong permissions would be applied. Rather than applying the permissions for the folder that contains the requested file, those of an ancestor folder would be applied.

The Web Server Folder Traversal vulnerability differs because it doesn't target CGI scripts and ISAPI extensions (although it could); it can map to a specific file, (cmd.exe for example), execute the file and run other valid commands located on the file system or execute simple HTTP commands such as GET and PUT. The danger associated with this vulnerability is tremendous.

Depending on the level of competence the intruder or attacker may have, this vulnerability exposes the computer system to a wide variety of attacks. A low-level attacker may be interested in web page defacement. This could be accomplished by coping and overwriting the default web page. This level of attacker may also have malicious intentions such as deleting server content. A mid-to-upper-level attacker may be interested in activity such as chat room hosting, IP hopping, or file storage. These activities are accomplished by copying a program such as NetCat or BackOrifice to the

vulnerable computer, then establishing a path that will allow the attacker unrestricted access to the vulnerable computer. A high-level attacker may be interested in activities such as espionage or information stealing. This type of attacker/intruder will typically leave no trace of his or her presence.

Armed with the information provided by the File Permission Canonicalization vulnerability, an anonymous person discovered that by using the UTF-8 Unicode to create a path to a particular command, he or she could execute the DIR command on an IIS. This information was posted to a forum for others to see and try.

Par Osterberg read the post concerning this possible vulnerability and sent an email to Rain Forest Puppy (rfp). Rfp is an individual/organization that exposes possible computer vulnerabilities. Rfp examined the claims of the vulnerability using IIS v.5.0 running on a Windows2000 operating system. Noticing that the example web site given by the post was foreign, rfp searched for sites hosting the default IIS v5.0 web page and using foreign Unicode fonts. All of the tested sites were vulnerable.

This prompted for a testing of IIS hosted by U.S. sites. Once the testing of U.S. sites were completed, none could be exploited using this vulnerability. Since foreign sites were affected and US sites were not, rfp concluded that the reasons must lie within the Unicode translation.

Rfp created a perl script that would check all 65,535 variations of the Unicode combination %c1%1c as noted on the Packetstorm post. When the check was complete, two variations resulted in a directory listing on the IIS - %c0%af and %c1%9c. The same two variations were then tested against IIS v4.0 and yielded the same results – a directory listing.

Rfp determined that the overlong Unicode representations for '/' and '\' are %c0%af and %c1%9c respectively. It was determined that IIS decodes Unicode after path checking, not before. Rfp contacted Microsoft to disclose their findings. Microsoft then issued a release "MS00-078" to warn of this situation. Microsoft also noted that the patch from the MS00-057 Bulletin, ("File Permission Canonicalization"), fixes this problem.

The technical overview of this vulnerability could be broken into three categories. Hypertext Transfer Protocol (HTTP) overview, this overview shows how commands are given to Web-Servers; UTF-8 UNICODE Overview, this overview shows how commands are converted between ASCII and Unicode; and Web Server Folder Traversal Overview, this overview show how the vulnerability is exploited.

HTTP services that reside on a Web-Server will accept input by default on TCP port 80. When a client connects on port 80, the Web-Server will accept requests for web pages or data in HTTP. The Web-Server performs action on the requests and returns data to the client. Each individual request for information between client and server is a single transaction and the TCP connection is closed after the Web-Server has responded. E.g.

Client sends first request on port 80 to Web-Server – Web-Server accepts, processes, and responds to first request then closes connection. Client sends second request on port 80 to Web-Server – Web-Server accepts, processes, and responds to second request then closes connection.

To test this process, individuals can telnet into a Web-Server on port 80 and issue a GET command. A "GET" command with no argument will return an error to the client and close the connection. A "GET /" command will return the Web-Server's default web page then close the connection. A "GET /valid_path/valid_filename" command will return the file then close the connection. An example would be, "GET /images/default.jpg". This command would display the bit code of default.jpg. A "GET /valid_path/invalid_filename" command will return an error then close the connection.

ISO/IEC 10646-1 defines a multi-octet character set called Universal Character Set (UCS). The UCS attempts to assign a unique name and number to each character and symbol found in every known language. Our common everyday character set, US-ASCII, is not a multi-octet character set. US-ASCII is represented by a single octet. Many computer applications that support US-ASCII are not compatible with multi-octet characters. UCS Transformation Formats (UTF) are used to circumvent this problem. UTF-8 describes a translation format that assigns US-ASCII to a single octet while providing multi-octet functionality to other character sets. UTF-8 describes a translation format that requires no translation between US-ASCII and UTF-8 Unicode.

The rules of UTF-8 state that if the original character is 7 bits or less, the character remains unchanged. It is merely preceded with a '0' (0xxxxxxx). An example would be the letter 'A'. The hexadecimal number 41 in US-ASCII represent 'A'. 41 in hexadecimal is 1000001 in binary. 1000001 are 7 bits long, which is less then 8 bits. UTF-8's representation would be 01000001 binary or 41 hexadecimal. Therefore, US-ASCII 41 is equivalent to UTF-8 Unicode 41 – "A".

UTF-8 rules state that a characters less then 12 bits but greater the 7 bits will be represented as 110xxxxx 10xxxxxx. If a character has a hexadecimal value of 10F, it would have a binary value of 100001111. 100001111 are 9 bits long. UTF-8's representation of this character set would be "11000100 10001111" binary or "C4 8F" hexadecimal. This number is arrived by adding the bits according to the rules stated by UTF-8. Adding 100001111 into 110xxxxx 10xxxxxx and filling in the blank (x) spaces with 0s.

110xxxxx 10xxxxxx	← UTF-8 Rule for 7 to 12 Bits
+ 100 001111	← Binary value of Hexadecimal character 10F
110xx100 10001111	← Result of the two values added together (less padding of 0s)
11000100 10001111	← Results of two values added together w/0s
C4 8F	← Hexadecimal value

One consideration to understand is that C4 F8 is not considered a multi-octet character and should not be confused as two single octet characters even though the binary octet starts with the number one. All single octet characters start with a zero. RFC 2279 section 6 "Security Considerations" addresses this specific topic. Microsoft failed to note this and as a result, introduced a vulnerability ("Web Server Folder Traversal") into its Internet Information Server software.

UTF-8 representation summary:

00-07 bits	0xxxxxxx
08-11 bits	110xxxxx 10xxxxxx
12-16 bits	1110xxxx 10xxxxxx 10xxxxxx
17-21 bits	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

Using the UTF-8 rules and the HTTP Overview together, an individual using telnet could (over port 80) send the following command: "%47%45%54 %57%57%57" which means, "GET WWW."

When completing an install of Microsoft's Web-Server software IIS, there are a few directories that are automatically installed. Two of the installed directories are "C:\inetpub" and "C:\inetpub\wwwroot". Although other directories are created during installation, they are not relevant to this discussion.

"C:\inetpub\wwwroot" is the default directory for the HTML WebPages of the server. "C:\inetpub\wwwroot\default.htm" is the default web page of the server. By default, an individual can issue the command "GET /" over TCP port 80 and receive the servers default web page.

An individual whose current path is "C:\inetpub\wwwroot" and using a DOS prompt could issue the following command: "type ..\..\autoexec.bat." This command would display the contents of the file "autoexec.bat" which is located on the root directory "C:\". The command "..\" indicates to navigate one directory level up. Therefore, the command "..\..\" indicates to navigate two directories levels up, (one level to exit the "wwwroot" directory and another level to exit the "inetpub" directory). This is referred as using relative paths.

However, an individual connecting to TCP port 80 on the IIS Web-Server that issues a similar command, ("GET ..\..\autoexec.bat"), would receive an error. IIS filters requests from a client to prevent relative pathnames and other operating system characters that attempt to redirect files access outside the Web-Server document tree.

An individual with and understanding of both concepts of UTF-8 Unicode translations and HTTP could circumvent the protective measures of IIS as it pertains to relative pathnames and file access. Using the Web Server Folder Traversal exploit, a request to redirect file access outside the Web-Server's document tree is disguised by

using overlong Unicode. An unpatched IIS improperly decodes Unicode at the wrong instance, after the path-checking routine is conducted, versus before.

Overlong Unicode results when a single octet US-ASCII character is improperly encoded as a dual octet UTF-8 character. The single octet US-ASCII character "A", as an example, is encoded by the following rule – 0xxxxxxx resulting in 01000001 binary or 41 hexadecimal. When erroneously encoding the single octet US-ASCII character "A" using the 110xxxxx 10xxxxxx rule, we would receive a binary number of 11000001 10000001 or C1 81 hexadecimal. The unpatched IIS will incorrectly decode %C1%81 as the letter "A", furthermore, it performs this incorrect decode after the path-checking routine.

The letter "A" is not very useful when navigating relative paths, however, using the same principle, an individual could use %c0%af which represents "/", or "%c1%9c" which represents "\" and navigate throughout the system to complete simple dos commands.

This could be accomplished by inserting the following string into the address portion of a client web browser:

"http://www.vulnerable.com/msadc/..%c0%af../..%c0%af../..%c0%af../winnt/system32/cmd.exe?/c+dir+..\..\..\..\"

"http://www.vulnerable.com"
Represents the address of the unpatched IIS.

"/msadc/"
Represents the starting point of your navigation. Most IIS installations are configured to point to the "msadc" directory. This directory is usually located at "C:\Program Files\Common files\msadc."

"..%c0%af/"
Represents the first relative path from "C:\Program Files\Common Files\msadc"; that path is "C:\Program Files\Common Files.

"..%c0%af/"
Represents the second relative path from "C:\Program Files\Common Files\msdac"; that path is "C:\Program Files"

"/winnt/system32/cmd.exe?"
Represents the command you intend to execute.

"c+dir+..\..\..\"
Represents the parameters of the executed command. "C" indicates to carry out the command specified by then stop. Dir indicates what command to execute. "..\..\..\..\" indicates what directory (relative to msadc) to list.

Users of IIS are automatically given access rights associated with the IUSR_machinename user account. The IUSR_machinename account is a member of the "everyone" and "users" groups by default. Therefore, any file on the same logical drive

and any web-accessible file that is accessible to these groups and be deleted, modified, or executed.

The following represents a timeline associated with this vulnerability:

10/10/2000	Message posted to Packetstorm
10/11/2000	Vulnerability tested (unsuccessfully) by other forum users
10/14/2000(est)	Message brought to the attention of Rain Forest Puppy
10/15-16/2000	Vulnerability brought to the attention of Microsoft
10/17/2000	Microsoft bulletin posted

DEFINITIONS

BACKORIFICE	Back Orifice is a tool consisting of two main pieces, a client application and a server application. The client application, running on one machine, can be used to monitor and control a second machine running the server application.
CANONICALIZATION	Canonicalization is the process by which various equivalent forms of a name can be resolved to a single, standard name - the so-called canonical name. For example, on a given machine, the names c:\dir\test.dat, test.dat, and ..\..\test.dat might all refer to the same file. Canonicalization is the process by which such names would be mapped to a name like c:\dir\test.dat.
CGI	Computer Gateway Interface
DIR command	Displays the names of files on a disk along with other information such as the time and date the file was changed, the size of the files, the number of files in a specific directory, the size of all files combined and the total amount of free space left on the drive.
HTTP	Hypertext Transfer Protocol
ISAPI	Internet Server Application Programming Interface
NETCAT	Simple utility which reads and writes data across network connections, using TCP or UDP protocol
URI	Uniform Resource Identifier. The generic set of all names/addresses that are short strings that refer to resources.
URL	Uniform Resource Locator. The set of URI schemes that have explicit instructions on how to access the resource on the internet.
UTF-8 Unicode	UTF-8 encodes UCS-2 or UCS-4 characters as a varying number of octets, where the number of octets, and the value of each, depend on the integer value assigned to the character in ISO/IEC 10646.

Works Cited

anonymous "IIS5 has a very big bug that let you execute arbitrary comma"
http://209.143.242.119/cgi-bin/cbmc/forums.cgi?authkey=anonymous&uname=anonymous&datopic=Windows&msgcheck=defined&gum=474&editoron=
(10 Oct 2000)

Gischer, Matt "Weekly Microsoft Security Digest 2000/10/16 to 2000/10/22"
http://www.securityportal.com/topnews/weekly/microsoft20001023.html
(22 Oct 2000)

Icsa.net "IIS Directory Traversal Vulnerability"
http://www.icsa.net/html/hypeorhot/iistraversal.shtml
(17 Oct 2000)

Kuhn, Markus "UTF-8 and Unicode FAQ for Unix/Linux"
http://www.cl.cam.ac.uk/~mgk25/unicode.html#utf-8
(06 Feb 2001)

Microsoft "Microsoft Security Bulletin (MS00-057): Frequently Asked Questions"
http://www.microsoft.com/technet/security/bulletin/fq00-057.asp
(10 Aug 2000)

Microsoft "Microsoft Security Bulletin (MS00-078)"
http://www.microsoft.com/technet/security/bulletin/ms00-078.asp
(17 Oct 2000)

Microsoft "Microsoft IIS 4.0 patch"
http://www.microsoft.com/ntserver/nts/downloads/critical/q269862/default.asp
(17 Oct 2000)

Microsoft "Microsoft IIS 5.0 patch"
http://www.microsoft.com/windows2000/downloads/critical/q269862/default.asp
(17 Oct 2000)

rain forest puppy "IIS %c1%1c bug"
http://www.wiretrip.net/rfp/p/doc.asp?id=57&iface=3

Yergeau, F. "UTF-8, a transformation format of ISO 10646."
http://www.utopia.com/talent/lpb/muddex/essay
(Jan 1998).